RoboEarth Web-Enabled and Knowledge-Based Active Perception

L. Riazuelo[1], M. Tenorth[2], D. Di Marco[3], M. Salas[1], L. Mösenlechner[4], L. Kunze[5], M. Beetz[2], J. D. Tardos[1], L. Montano[1], J. M. M. Montiel[1]

***Abstract*— In this paper we explore how a visual SLAM system and a robot knowledge base can mutually benefit from each other. The object recognition and mapping methods are used for grounding abstract knowledge and for creating a semantically annotated environment map that is available for reasoning. The knowledge base allows to reason about which object types are to be expected while exploring an environment and where to search for novel objects given a partial environment map. Prior information like task descriptions and object models is loaded from RoboEarth, a web-based knowledge base for exchanging knowledge between robots, and the created maps are again uploaded to RoboEarth. We show that by exploiting knowledge about common objects in a room and about the co-occurrence of objects, both efficacy and efficiency of the perception can be boosted.**

I. INTRODUCTION

The abilities to efficiently create semantic environment models and to use these models intelligently to locate objects will become increasingly important as more and more robots enter human living and working environments. To successfully operate in such environments, robots will have to face the *open-world challenge*, i.e. they will need to be able to handle high numbers of (novel) objects that are located in various places on top of or inside furniture, and they need to quickly become acquainted with novel environments.

Competently performing these tasks poses several challenges for today's robots, for example: How can the visual perception system handle large numbers of object models without slowing down the recognition or detecting more false positives? How can a robot efficiently explore an environment to create a semantic map of objects? Which are the most important objects to look out for? How can the robot exploit common-sense knowledge to guide its search for novel objects? How can it profit from information other robots have already collected? We believe that finding solutions to these problems will be crucial to scale object search tasks from restricted and well-known laboratory environments to more open and diverse scenes.

In this paper, we investigate a web-enabled and knowledge-based approach to this problem. Our robots have access to the web-scale RoboEarth knowledge base [24] that provides them with formally represented task descriptions, different kinds of environment maps and both semantic and geometric object models. Through RoboEarth, robots can share information by uploading maps of environments they have explored or descriptions of tasks they have learned. By intelligently selecting only that information that is needed for the current task from this web-based knowledge base, robots can keep their local knowledge bases and object model databases small and efficient, while having much larger information resources in the background.

All pieces of information in RoboEarth are semantically annotated, i.e. they are described in a formal, logical language [23] and linked to an ontology. This enables the robot to perform logical inference to verify that all capabilities that are required for a task are available, to compute the most likely locations for an object, and to query for all objects of a certain type in the environment. Using inference on its background knowledge, the robot can decide which are the most likely objects in a room (and only download their models to its local database), can compute where novel objects are likely to be found (and guide the search accordingly), and thereby become more efficient in performing its tasks.

In order to apply the abstract knowledge to operation in the real world, it needs to be *grounded* [8] in the robot's perception system and its knowledge about the environment. In this paper, we propose to link the knowledge base with a visual SLAM system that provides accurate and continuous asynchronous perception and is integrated with an object recognition module that identifies objects based on a local database of object models. The main contributions of this paper are (1) the integrated system synergetically combining a SLAM and object recognition system with a knowledge base; (2) techniques for using prior knowledge to select object models for exploration; and (3) methods for guiding object search by exploiting background knowledge about objects and a partial semantic map.

The remainder of the paper is organized as follows: We start with an overview of related work on searching objects and explain the structure of our system and the two main tasks it performs: guided exploration and knowledge-based object search. We then present the system components in more detail, describe the experiments we have performed, and finish with our conclusions.

II. RELATED WORK

Structured object search and reasoning about likely object locations have been an active research topic over the past

[1]L. Riazuelo, M. Salas, J. D. Tardos, L. Montano, J. M. M. Montiel are with Aragón Institute of Engineering Research (I3A), University of Zaragoza, Spain. {riazuelo, msalasg, tardos, montano, josemari}@unizar.es

[2]M. Tenorth, M. Beetz are with Universität Bremen, Germany. {tenorth, beetz}@cs.uni-bremen.de

[3]D. Di Marco is with Universität Stuttgart, Germany. dimarco@ipvs.uni-stuttgart.de

[4]L. Mösenlechner is with Technische Universität München, Germany. {moesenle}@cs.tum.edu

[5]L. Kunze is with Intelligent Robotics Lab, University of Birmingham, United Kingdom. {l.kunzel}@cs.bham.ac.uk

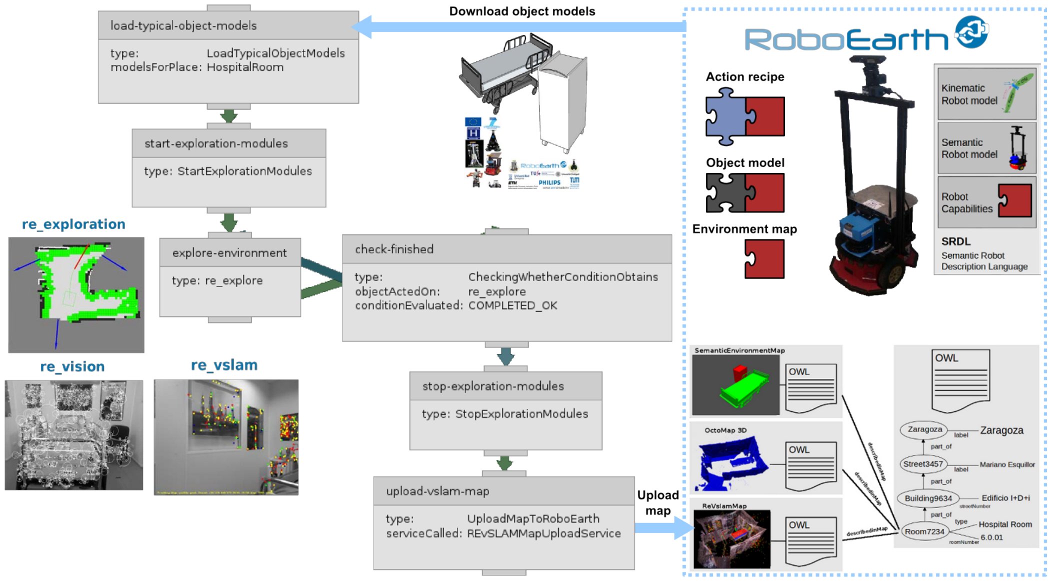

Fig. 1. System overview. The robot downloads information from the RoboEarth knowledge base (right), executes the task based on the downloaded specification (grey blocks), explores the environment, detects objects and creates a semantic environment map. The resulting can then be uploaded to RoboEarth to share it with other robots.

years. Much work explored vision-based methods to search for objects in a top-down manner based on saliency and visual attention mechanisms [17], [5], [21]. Zhou et al. [27] use information from Web-based common-sense databases and search engines to compute object–room co-occurrence probabilities. Kunze et al. propose a utility-based approach for object search that particularly focuses on the decision of which location to search first [10]. The approach by Aydemir et al. [2] is similar to ours in that they also use landmark objects to guide the search for smaller objects inside or on top of the landmarks. While they focus on the probabilistic formulation of the search procedure as a Markov Decision Process, we explore a knowledge-based strategy that exploits formal knowledge about object types, their (likely) spatial relations, and their shape and appearance.

III. Overview

Figure 1 gives an overview of the main components of the proposed system on the example of the semantic mapping task. We assume that the RoboEarth knowledge base (right block) contains a task description (called "action recipe") for the semantic mapping tasks and models for common objects. In this paper, we focus on two action recipes for perception tasks: semantic mapping of an unknown environment and active search for an object. The second recipe for active search exploits knowledge about the locations of already detected objects in the room that act as landmark objects.

Each piece of information in RoboEarth is annotated with a formal description of the requirements a robot has to fulfill in order to use it. This description (depicted as colored puzzle pieces) is matched against a formal model of the robot's capabilities using the Semantic Robot Description Language (SRDL) [11] to make sure that the information can be used by the robot. Based on the background knowledge about which objects are likely to be encountered in which kinds of rooms, RoboEarth infers a set of object models that will be recognized during the exploration. The downloaded object models are sent to the visual SLAM and object recognition components (Section V-C) that are able to insert recognized objects into the map in real time. The object models are in a local sub-database composed of about a dozen models, which ensures fast recognition and a low false-positive rate. The recognized objects are added to the SLAM map and serve as landmark objects when searching for objects later on.

After download, a robot plan is generated from the action recipe (Section IV), and the robot executes the task according to the structure defined in the recipe. It explores the environment using a frontier-based algorithm (Section V-B) and creates a visual SLAM map including the recognized objects (Section V-C). After the exploration has finished, the robot exports the map in the formal RoboEarth language and uploads it to the RoboEarth knowledge base.

IV. Action Recipes for Active Perception Tasks

Action Recipes [23] specify a task on a platform- and environment-independent level. They are formulated in the Web Ontology Language (OWL) in terms of action classes defined in a common ontology. Recipes describe the action to be executed, objects to interact with, as well as constraints on the execution flow. In comparison to a robot execution plan, they are formulated on an abstract level (see e.g. Figure 1). This enables sharing recipes among robots with different hardware configurations. However, due to the level of abstraction, this representation is not viable to be used as input to an interpreter for task execution, and the information missing in the recipe needs to be included first. Our system thus has a set of manually written functions that generate partial execution plans for different OWL action concepts. They can be thought of as templates for the final plan steps, which are completed using the description of the robot and the environment. For example, the code generating function for the visibility reasoning step (Section VI) looks up the pose of the camera used for object detection relative to the robot base in the robot's SRDL description. The code generating functions are stored in the RoboEarth database and annotated with an OWL description that defines the OWL classes and robot SRDL for which it is able to generate executable code. For each action described in the recipe, the system looks up possible code generator functions in the RoboEarth database and checks which are usable for the given robot platform by matching the SRDL capabilities. In case multiple code generator functions are found, it chooses the one with minimal Rada distance [4] to the action concept under consideration. If the result is still ambiguous, the human operator is asked. The code generator functions then get the robot platform used and the environment as parameters and extract required parameters themselves. We chose the CRAM plan language [3] as notation for executable plans.

In this paper, we mainly investigate two robot tasks: semantic mapping of an environment and knowledge-based object search. Both tasks have been described in terms of a RoboEarth action recipe. The *SemanticMapping Recipe* enables a robot to build a semantic map from scratch and to store the result as a RoboEarth environment. Before the task, the knowledge base infers a set of landmark objects which are typically found in the type of room that is being explored. A semantic map including these landmark objects is useful in future tasks to guide the search for small objects which are hard to detect if the robot is not close to them. The small set of landmark objects allows real-time map building and increases the robustness with respect to recognition errors because it contains only those objects that are likely to be in the room. To compute the exploration trajectory, we apply a frontier exploration algorithm which uses path planning and reactive navigation techniques for safe navigation. Once the exploration is finished, the semantic map is uploaded to RoboEarth for future use.

The *ObjectSearch Recipe* exploits the semantic map built in an earlier exploration run, not necessarily done by the same robot. The knowledge base infers the potential locations from where the selected object is probable to be detected (see Section VI for details about the inference). The robot safely navigates directly towards the computed locations and starts a local search until the object is found. The found objects are included within the semantic map, which is uploaded to the RoboEarth knowledge base.

V. Robot Capabilities for Active Perception

An action recipe specifies the set of capabilities needed for its execution. A robot *capability* is defined as the ability to perform a certain action. A robot effectively has a capability if it has the necessary hardware and low-level software *components* that implement this capability. The set of available capabilities also influences the structure of the robot plan that is generated from the action recipe. For the case of the active perception tasks described in this work, the needed robot capabilities are depicted in Figure 2. The following subsections describe these capabilities and the corresponding components.

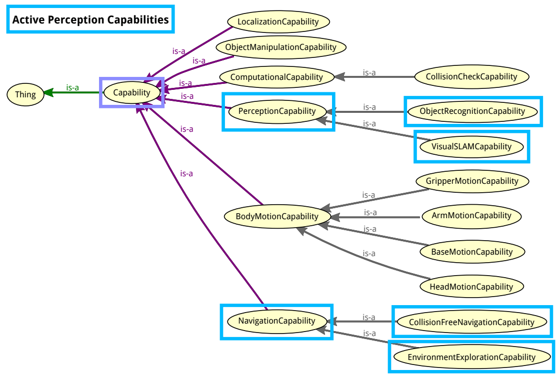

Fig. 2. A summary of the SRDL robot capabilities ontology including the active perception capabilities.

A. Navigation and Exploration Capabilities

The *NavigationCapability* class comprises the following navigation capabilities:

- *EnvironmentExplorationCapability*: Ability to navigate in an unknown environment in order to build a navigation map. This capability is implemented by an exploration algorithm that is represented as an instance of the class *EnvironmentExploration*.
- *CollisionFreeNavigationCapability*: Ability to safely navigate to a goal. This capability is implemented by obstacle avoidance and trajectory planning algorithms which are described as instances of the class *NavigationComponent*.

The *PerceptionCapability* class groups the following perception-related capabilities:

- *VisualSLAMCapability*: Ability to build a map composed of point features, objects and a 3D grid cell map. This capability is implemented by a SLAM algorithm of the type *VisualSlamMappingComponent*.
- *ObjectRecognitionCapability*: Ability to recognize objects and provide an initial estimate of its location. This capability is implemented by a visual recognition algorithm of type *ObjectRecognitionComponent*.

B. Navigation and Exploration Components

The *Navigation component* implements robot trajectory planning and collision avoidance algorithms for safe navigation towards the goal location. Whilst the robot moves, a 2D map is built and the robot is continuously localized in it. This component is implemented by the following algorithms:

1) An integrated path planning and obstacle avoidance method. The global navigation plan is locally modified by the reactive navigation, which is in charge of computing the motion command. The planning technique is based on a A*-type algorithm [13]. As obstacle avoidance method we have applied ORM [15] adapted for differential drive robots due to its performance on dense, complex and cluttered environments.

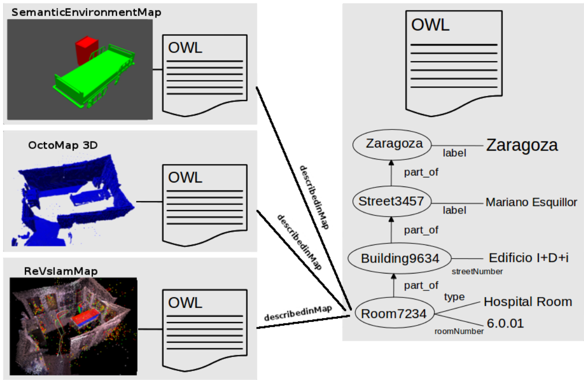

Fig. 3. Output of the semantic visual map stored according to RoboEarth standards for environment model representation. The right box encodes meta-information about the map to be used for searching the RoboEarth database. The top left represents the semantic map composed of objects. The middle left represents the occupancy grid, the bottom left the visual SLAM map.

2) A Rao-Blackwellized particle filter SLAM [6] to estimate the robot location and the 2D navigation map, computed from 2D laser rangefinder readings.

The purpose of the *EnvironmentExploration* component is to provide the capability to explore a room for the first time. It is very unlikely to build a full map of the room just from the first sensor reading, so the main issue is to compute on-line the robot locations from where to perceive unexplored room regions. The exploration uses a frontier-based approach [26] to compute the robot exploration goals. The robot moves to an unexplored area while avoiding obstacles using the *Navigation component* and adds the new information to its map. The exploration ends when the map contains no more accessible frontiers.

C. Perception Components

The goal of the perception components is to provide environment maps, see Figure 3. RoboEarth classes for storing environments are:

- *SemanticEnvironmentMap:* These maps are described in OWL and consist of detected objects in the environment. Objects are described as instances of the respective object classes in the ontology, which allows the application of logical inference methods to the spatial configuration of objects.
- *OctoMap:* 3D occupancy grid map. The Octomap algorithm [25] computes a 3D occupancy grid for the room from an RGB-D sensor jointly with the camera trajectory. This map can be reused to generate 2D maps for navigation.
- *ReVslamMap:* Raw storage of visual maps for localization that are built from the sole input of an RGB-D camera. Along with the creation of the room map, the camera trajectory is estimated. For visual SLAM, we have used the C2TAM [18] SLAM algorithm which is based on PTAM [9]. Additionally, objects in the local sub-database are detected in the images and inserted in the map. The final result is a semantic map of the observed room [19]. This component is used both in the *SemanticMapping* recipe and in the *ObjectSearch* recipe.

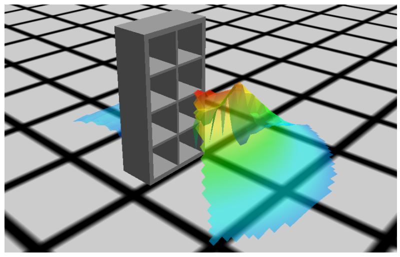

Fig. 4. Visibility costmap computed from the semantic environment map, the semantic robot model and geometric object models downloaded from RoboEarth.

VI. Reasoning about Object Locations

In order to successfully find an object in the environment, a robot must answer the questions "Where is the object likely to be?" and "Where do I need to go in order to see it?".

1) Inferring likely object positions: We employ knowledge that has been extracted from the OMICS common-sense database [7] and converted into the representation used in the robot's knowledge base [12] to compute likely object positions. The OMICS database holds tuples of objects and their locations in the form $(object, location)$. The number of times a relation is contained in OMICS can be used to approximate the likelihood that an object O can be found at a location LOC:

$$P(O|LOC) = count(O, LOC)/count(LOC) \quad (1)$$

where $count$ is the number of database entries. The value of $P(LOC|O)$ is calculated from the above model using Bayes rule. To retrieve the location with the highest probability we simply apply the *argmax* operator

$$\underset{LOC \in Locations}{\operatorname{argmax}} \quad P(LOC|O) \quad (2)$$

The resulting models allow queries for the locations of objects given by corresponding landmark objects. These object classes can be grounded in the robots semantic environment map to determine their positions.

2) Computing robot poses using visibility reasoning: Based on the semantic map (that contains known object instances in the environment) and CAD models of these objects previously downloaded from RoboEarth, the system computes a visibility costmap describing from which robot poses the object is likely to be visible [16]. Especially for objects that are inside a cabinet or shelf, occlusions by the surrounding objects need to be taken into account when planning a pose for the robot. To compute the costmap, the system renders the scene from the viewpoint of the inferred object location and computes the amount of the object that is visible from all grid-cells in the costmap (Figure 4).

VII. Experiments

To validate the web-enabled and knowledge-based active perception, both action recipes presented before have been tested on a real Pioneer P3-DX robot platform operating in a

Fig. 5. Bed and cabinet detected during the semantic mapping action recipe.

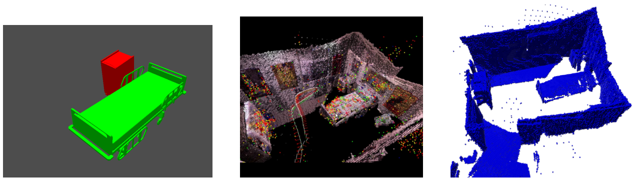

Fig. 6. (From left to right). Semantic map composed of a bed and a cabinet, map of visual features and octomap generated during the action recipe.

mock-up hospital room environment. In addition, we present experiments in simulation that show how this system operates on different robots and in different environments. Finally, we provide results[1] that show the performance improvement obtained by using the proposed method.

A. Real-world experiments

Regarding the hardware and software components for visual semantic mapping, we have used a Kinect RGB-D sensor and, for navigation purposes, the robot is equipped with a Sick 2D laserscanner and odometry sensors. It also incorporates the *move_base* ROS stack with the ORM obstacle avoidance method and GMapping package. The ROS RoboEarth stack [1] and the KnowRob knowledge base [22] provide the inference methods used.

The following scenario has been investigated: A robot in a hospital room has to find a bottle to be served to a patient. The robot initially does not know the location of the bottle in the room. A naïve solution would be to exhaustively search the room to find the bottle, but it is a costly process. To improve efficiency, we propose a knowledge-based search strategy based on the action recipes mentioned earlier. The *SemanticMapping* recipe enables the robot to efficiently create a semantic map of the environment during the exploration of the room. Before the task, the knowledge base infers that the bed and the cabinet are likely landmark objects, and the corresponding object models are downloaded into the model sub-database on the robot. A customized plan for an efficient exploration is generated for the robot based on the recipe. The robot executes the tailored plan and starts to explore the unknown environment until it obtains a complete map of the room. While the robot is exploring the environment, the perception component detects objects in the sub-database. Figure 5 shows two snapshots of the object recognition component output corresponding to two object recognition events. The generated semantic map is then uploaded to RoboEarth (Figure 6).

[1]A video of the experiments can be found at `http://robots.unizar.es/data/videos/roboearth/activePerception.mp4`

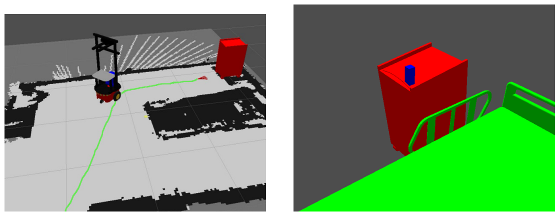

Fig. 7. Left, in red robot location for optimal perception, in green the planned trajectory to those locations. Right, the final semantic map including the just found bottle on top of the cabinet (detected during the creation of the original map).

The second recipe allows guided search for objects based on a partial semantic map of the environment – in our case a map containing a bed and a cabinet that has been built in a previous exploration run. The object location inference determines the cabinet as likely location for the bottle and computes several reachable robot locations from where the bottle is probably detected based on the current scene layout and the free space. A CRAM plan is generated for the Pioneer P3-DX robot, and a suitable 2D occupancy map for navigation is computed based on the 3D octomap and the semantic robot description. Overall, RoboEarth provided the plan, the set of recognition models and the maps for safe robot navigation to the robot.

The robot navigates to the computed locations until the object is found (Figure 7). Once the object is located, the updated map containing the object is uploaded. Figure 7 shows the semantic map including the objects known a priori (bed and cabinet) and the new one (bottle). The visual semantic mapping also updates its map and integrates the newly gathered information.

B. Simulation Experiments

To show the applicability of the system on different robot hardware and different environments, we also did some simulation experiments. We used both the Amigo robot, a service robot prototype with a holonomic base [14], and the previously described Pioneer.

Figure 8 shows the travelled path for both robots in two different simulated environments in blue color. The paths mirror the difference in locomotion; the Amigo robot is able to maneuver more efficiently in the tight spaces than the non-holonomic Pioneer. For the visibility reasoning, the pose of the camera relative to the respective robot base is required. This value was inferred from the SRDL descriptions of the robots during plan generation.

C. Performance Improvements

To show the improvement made by the proposed system, the same experiments have been done without using the information from the semantic map. In this case, the robot has to perform an exhaustive search of the entire environment to find the small object. To perform this search, we used the art gallery algorithm [20] that gives us the minimum number of positions which can cover an environment. Figure 9 shows the distribution of the positions needed for full coverage of the entire environment, which are between 20-100. The presented inference methods exploit the semantic information of the

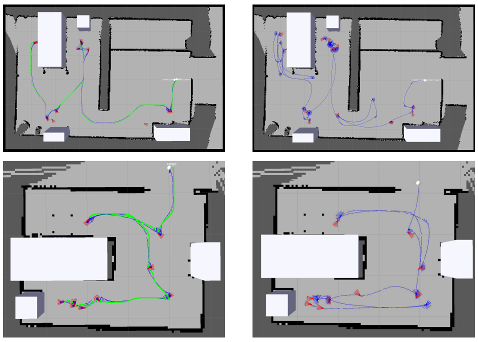

Fig. 8. Travelled path (blue) in simulated object search for Amigo (left) and Pioneer (right) robots in two different simulation environments.

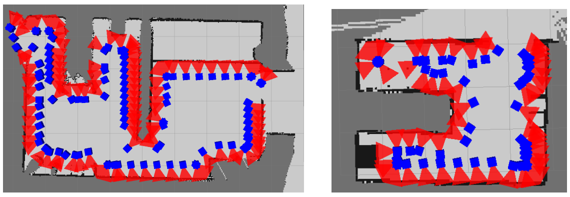

Fig. 9. Exhaustive search of a small object. Blue squares are the potential visibility positions, and red areas are the field of view of the camera in order to detect a small object.

environment (Figure 8) and are able to reduce this number to between 9 to 15 locations (with different orientations) where the small object could be found. This leads to a substantial improvement in efficiency and search time.

VIII. Conclusions and Future Work

A basic robot with state-of-the-art navigation and perception capabilities is not able to efficiently explore and actively search for an object. We have shown that the robot performance in active perception is boosted by web services provided by RoboEarth. Our experiments provide experimental support for the initial claim.

We showed how the RoboEarth system handles the robot diversity because it is able to deliver an execution plan that is customized for the current robot and current environment. RoboEarth also provides the robot with a selection of only models that are relevant for the current task. The rather small number of required models allows to obtain perception with high precision and recall in real time. In future work, we plan to extend the range of environments, robots and sensors that can benefit from the RoboEarth boost.

IX. Acknowledgements

The research was partly funded by the EU FP7 project RoboEarth (grant number 248942), and by the Spanish Government grants DPI2012-32168 and DPI2012-32100.

References

[1] ROS packages for interfacing with RoboEarth. http://www.ros.org/wiki/roboearth_stack. Accessed: 2013-02-19.

[2] A. Aydemir, K. Sjoo, J. Folkesson, A. Pronobis, and P. Jensfelt. Search in the real world: Active visual object search based on spatial relations. In *IEEE Int. Conf. on Robotics and Automation (ICRA)*, pages 2818–2824, 2011.

[3] M. Beetz, L. Mösenlechner, and M. Tenorth. CRAM – A Cognitive Robot Abstract Machine for Everyday Manipulation in Human Environments. In *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*. 2010.

[4] E. Blanchard, M. Harzallah, H. Briand, and P. Kuntz. A typology of ontology-based semantic measures. In *EMOI-INTEROP workshop at the 17th Conf. on Advanced Information Systems Engineering*, Porto, Portugal, 2005.

[5] S. Ekvall and D. Kragic. Receptive field cooccurrence histograms for object detection. In *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems(IROS).*, pages 84–89, 2005.

[6] G. Grisetti, C. Stachniss, and W. Burgard. Improved techniques for grid mapping with rao-blackwellized particle filters. *IEEE Transactions on Robotics*, 23, 2007.

[7] R. Gupta and M. J. Kochenderfer. Common sense data acquisition for indoor mobile robots. In *Nineteenth National Conf. on Artificial Intelligence (AAAI-04*, pages 605–610, 2004.

[8] S. Harnad. The symbol grounding problem. *Physica D*, 42:335–346, 1990.

[9] G. Klein and D. Murray. Parallel tracking and mapping for small ar workspaces. In *Mixed and Augmented Reality, 2007. ISMAR 2007. 6th IEEE and ACM Int. Symposium on*, pages 225–234, 2007.

[10] L. Kunze, M. Beetz, M. Saito, H. Azuma, K. Okada, and M. Inaba. Searching objects in large-scale indoor environments: A decision-thereotic approach. In *IEEE Int. Conf. on Robotics and Automation (ICRA)*, St. Paul, MN, USA, 2012.

[11] L. Kunze, T. Roehm, and M. Beetz. Towards semantic robot description languages. In *IEEE Int. Conf. on Robotics and Automation (ICRA)*, pages 5589–5595, Shanghai, China, 2011.

[12] L. Kunze, M. Tenorth, and M. Beetz. Putting People's Common Sense into Knowledge Bases of Household Robots. In *33rd Annual German Conf. on Artificial Intelligence (KI 2010)*, pages 151–159, Karlsruhe, Germany, 2010. Springer.

[13] J.-C. Latombe. *Robot Motion Planning*. Kluwer Academic Publishers, Norwell, MA, USA, 1991.

[14] J. Lunenburg, S. van den Dries, J. Elfring, R. Janssen, J. Sandee, and M. van de Molengraft. Tech united eindhoven team description 2012. In *RoboCup Team Description Papers 2012*, 2012.

[15] J. Minguez. The obstacle-restriction method (orm) for robot obstacle avoidance in difficult environments. In *Proc. of the IEEE Int. Conf. on Intelligent Robots and Systems (IROS)*, 2005.

[16] L. Mösenlechner and M. Beetz. Parameterizing Actions to have the Appropriate Effects. In *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, San Francisco, CA, USA, 2011.

[17] A. Oliva, A. Torralba, M. S. Castelhano, and J. M. Henderson. Top-down control of visual attention in object detection. In *Int. Conf. on Image Processing (ICIP).*, volume 1, pages I–253, 2003.

[18] L. Riazuelo, J. Civera, and J. M. M. Montiel. C2TAM: A Cloud framework for Cooperative Tracking And Mapping. *Robotics and Autonomous Systems*, 2013. Under review.

[19] M. Salas and J. M. Martínez Montiel. Object visual SLAM. In *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, 2013. Submitted.

[20] T. Shermer. Recent results in art galleries. *Proc. IEEE*, 80(9):1384 – 1399, 1992.

[21] K. Shubina and J. K. Tsotsos. Visual search for an object in a 3d environment using a mobile robot. *Computer Vision and Image Understanding*, 114(5):535–547, 2010.

[22] M. Tenorth and M. Beetz. KnowRob – A Knowledge Processing Infrastructure for Cognition-enabled Robots. Part 1: The KnowRob System. *Int. Journal of Robotics Research*, 32(5):566 – 590, 2013.

[23] M. Tenorth, A. C. Perzylo, R. Lafrenz, and M. Beetz. Representation and Exchange of Knowledge about Actions, Objects, and Environments in the RoboEarth Framework. *IEEE Transactions on Automation Science and Engineering (T-ASE)*, 10(3):643–651, 2013.

[24] M. Waibel, M. Beetz, R. D'Andrea, R. Janssen, M. Tenorth, J. Civera, J. Elfring, D. Gálvez-López, K. Haussermann, J. Montiel, A. Perzylo, B. Schiele, O. Zweigle, and R. van de Molengraft. Roboearth - a world wide web for robots. *IEEE Robotics and Automation Magazine*, 18(2):69–82, 2011.

[25] K. M. Wurm, A. Hornung, M. Bennewitz, C. Stachniss, and W. Burgard. OctoMap: A probabilistic, flexible, and compact 3D map representation for robotic systems. In *Proc. of the ICRA 2010 Workshop on Best Practice in 3D Perception and Modeling for Mobile Manipulation*, Anchorage, AK, USA, 2010.

[26] B. Yamauchi. A frontier-based approach for autonomous exploration. In *Proc. of the IEEE Int. Symposium on Computational Intelligence, Robotics and Automation*, pages 146–151, 1997.

[27] K. Zhou, M. Zillich, H. Zender, and M. Vincze. Web mining driven object locality knowledge acquisition for efficient robot behavior. In *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, pages 3962–3969, 2012.